GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Parallel Programming for Multi-Core Architectures, PG_00064510						
Field of study	Informatics						
Date of commencement of studies	February 2027		Academic year of realisation of subject	2027/2028			
Education level	second-cycle studies		Subject group	Optional subject group Specialty subject group Subject group related to scientific research in the field of study			
Mode of study	Full-time studies		Mode of delivery	at the university			
Year of study	1		Language of instruction	English			
Semester of study	2		ECTS credits	4.0			
Learning profile	general academic profile		Assessment form	exam			
Conducting unit	Department of Computer Architecture -> Faculty of Electronics Telecommunications and Informatics -> Faculties of Gdańsk University of Technology						
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. inż. Paweł Czarnul					
	Teachers	dr hab. inż. Zdzisław Czarnul					
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	15.0	15.0	0.0	60
	E-learning hours included: 0.0						
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan		Participation in consultation hours	Self-study		SUM
	Number of study hours	60		9.0	31.0		100
Subject objectives	learning techniques of parallel programming and APIs allowing use of modern manycore platforms						

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K7_U04] can apply knowledge of programming methods and techniques as well as select and apply appropriate programming methods and tools in computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study, making assessment and critical analysis of the prepared software as well as a synthesis and creative interpretation of information presented with it	the student is able to select appropriate APIs and methods to optimise applications on multi-core systems	[SU1] Assessment of task fulfilment
[K7_W04] knows and understands, to an increased extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or other elements or programmable devices specific to the field of study, and organization of work of systems using computers or such devices	student knows basic rules and techniques of multithreaded programming for multi-core architectures	[SW1] Assessment of factual knowledge
[K7_U12] is able, to an increased extent, to analyze the operation of components and systems related to the field of study, as well as to measure their parameters and study their technical characteristics, and to plan and carry out experiments related to the field of study, including computer simulations, interpret the obtained results and draw conclusions	knows how to analyze and profile runs of parallel applications	[SU1] Assessment of task fulfilment
[K7_W10] knows and understands, to an increased extent, the basic processes occurring in the life cycle of equipment, objects and technical systems, as well as methods of supporting processes and functions, specific to the field of study	knows processes and dependencies concerning execution of parallel applications in multi-core systems	[SW1] Assessment of factual knowledge

Subject contents

Course content – lecture

1. Passing criteria
2. Current HPC systems
3. Goals of parallel programming
4. GPU as a parallel compute device
5. Data decomposition
6. Data parallel algorithms
7. CUDA programming model
8. GPU architecture
9. Threads in CUDA
10. Memory access in CUDA
11. Optimizations using CUDA
12. Using many GPUs
13. Application debugging
14. Unified Memory
15. OpenCL for GPUs/CPUs
16. Multicore CPUs
17. Many/multicore architectures
18. OpenMP
19. Offload, native, symmetric modes
20. Optimization (load balancing, synchronization)
21. Parallelization models for various paradigms in OpenMP.
22. Vectorization
23. False sharing
24. Thread affinity
25. Synchronization
26. Optimization divide-and-conquer
27. Optimization computing similarity of vectors
28. CPU+GPU programming

<table>
<tr><td>Prerequisites and co-requisites</td><td>basic knowledge of parallel programming

C programming knowledge</td></tr>
<tr><td>Assessment methods and criteria</td><td>
<table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>laboratories</td><td>50.0%</td><td>25.0%</td></tr>
<tr><td>project</td><td>50.0%</td><td>25.0%</td></tr>
<tr><td>exam</td><td>50.0%</td><td>30.0%</td></tr>
<tr><td>colloquium 1 + 2</td><td>50.0%</td><td>20.0%</td></tr>
</table>
</td></tr>
<tr><td>Recommended reading</td><td>
<table>
<tr><td>Basic literature</td><td>[1] Pawel Czarnul. Parallel Programming for Modern High Performance Computing Systems. Taylor & Francis. 2018 ISBN 9781138305953

[2] CUDA C programming guide. NVIDIA

[3] OpenMP specification

[4] OpenCL specification</td></tr>
<tr><td>Supplementary literature</td><td>CUDA documentation - NVIDIA presentations</td></tr>
<tr><td>eResources addresses</td><td></td></tr>
</table>
</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td></td></tr>
<tr><td>Practical activites within the subject</td><td>Not applicable</td></tr>
</table>

Document generated electronically. Does not require a seal or signature.